Software Design Document for the Land Information System: Data Management

Submitted under Task Agreement GSFC-CT-2

Cooperative Agreement Notice (CAN) CAN-00OES-01

Increasing Interoperability and Performance of Grand Challenge Applications in the Earth, Space, Life, and Microgravity Sciences

Version 2.1

Revision history:

Version	***Summary of Changes***	***Date***
1.0	Initial release.	8/13/02
2.0	Updated version	1/10/03
2.1	Revision on CT's feedback	3/14/03

Table of Contents

List of Figures

List of Tables

1. Introduction

The Land Information System (LIS) is designed to perform land surface simulation and data assimilation on parallelized computing platforms, at very high spatial resolutions (up to ~1km) and in near real-time. It imports a continuous flow of atmospheric forcing data and a collection of land surface parameter datasets, and produces a huge amount of land surface data to satisfy the needs of diverse users. Such an operation poses many challenges to the data handling functionality of LIS, and requires a highly reliable and efficient data management design.

This document designs the LIS data management. Specifically, the design covers the following five functional areas,

- End-to-end data flow
- Data retrieval, distribution and storage
- Data analysis capabilities, including interpolation, reprojection, sub-setting, and file format conversion
- Link to the user interface
- Interoperability through ALMA and ESMF compliance

The goal of the data management design is to have a system which will ensure smooth end-to-end data flow for LIS' distributed and near real-time operations, handle huge amount of data efficiently, provide useful data processing capabilities, be easily accessed, and can couple with other Earth system models through the standard ALMA data format and ESMF interfaces.

This document is organized as follows. Section 2 presents the global data flow design and traffic estimation at the network level. Section 3 and 4 document the design details of LIS input data and output data, respectively. Section 4 is also concerned with the connection between the output archive and the user interface. Section 5 deals with interoperability considerations of the data management aspect.

2. Data Flow Design and Data Volume Estimation

2.1 Overview of LIS Data Flow

LIS deals with three sets of globally gridded data: atmospheric forcing data, land surface parameter data, and output production data. LIS uses the atmospheric forcing data to drive the time evolution of the land surface models at each grid/tile unit, defined by the land surface parameters. The land surface simulation is performed by each land surface model's dynamical and physical equations, forced by the atmospheric data. The simulated results, a set of land surface and atmospheric related variables, constitute the LIS products, the output data.

Figure 1 shows LIS logical data flow on the LIS cluster platform. On SGI Origin platforms, the flow is the same, except that local disk operations on SGI will take the role of the IO nodes on the cluster, and the compute nodes' tasks on the cluster are performed by processors with shared-memory.

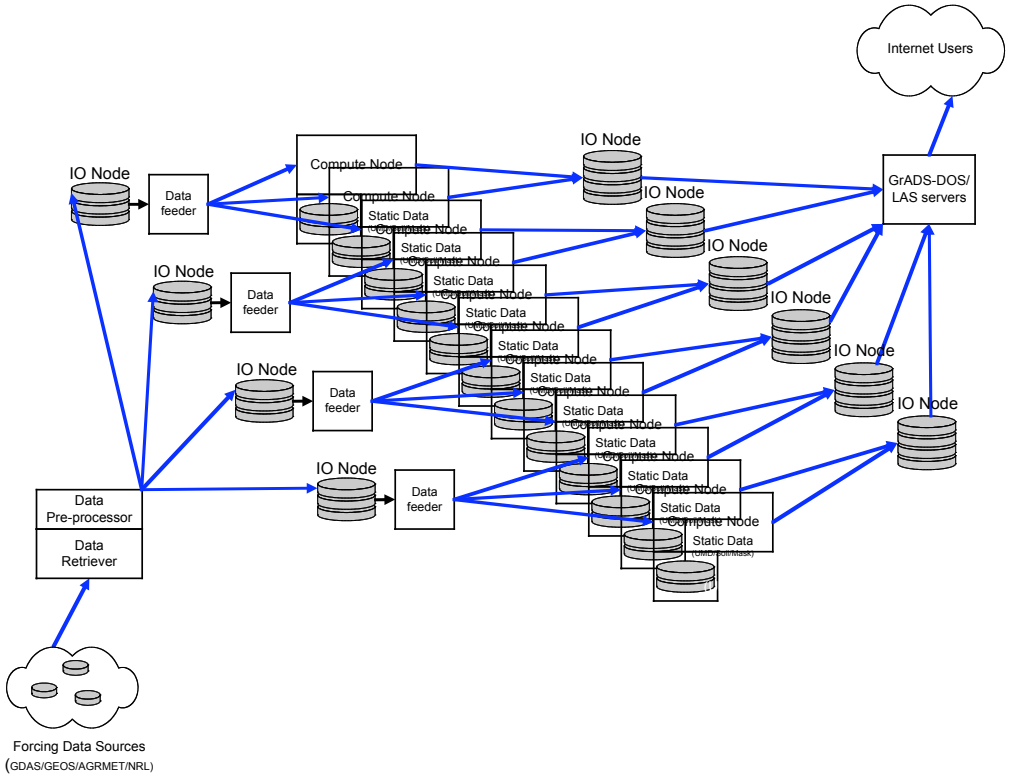

Figure 1: LIS global logical data flow and storage location on the LIS Linux cluster. Both the forcing data and the output data are stored on the IO nodes, while an identical copy of the land surface parameter data are stored on each compute node, since there parameter data are mostly static. On SGI's shared memory platforms, the flow is similar, except the functions of the IO nodes will be provided by local hard disks, and the compute nodes will be replaced by compute processors

As shown in Figure 1, the LIS end-to-end data flow involves two related areas: data storage on hard disks and data traffic over the network. The design of the data management has to make sure there are no bottlenecks in either area. In Section 2.2 below we will estimate the disk data volume, and in 2.3 we will analyze network traffic requirements.

2.2 Data Storage Convention

The primary LIS data storage file format will be binary and GRIB. Binary format will be used whenever access performance is critical, and GRIB will be used mainly for storage efficiency. Both the binary and GRIB data will be saved in the sequence which allows direct access by GrADS and GrADS-DODS server without reformatting.

Data saved in binary files are ordered as big endian, to maintain backward compatibility with LDAS code.

2.3 Disk Data Volume Estimation

Table 1, 2 and 3 listed all the data files and specifications. As described in Section 2.1, these files will supply LIS with atmospheric forcing, land surface parameters, initial and boundary conditions required for the land surface model runs, and will store the LIS output to serve users. For e.g, the atmospheric forcing data translate to variables such as total precipitation convective precipitation, downward shortwave and longwave radiation, near surface air temperature, near surface specific humidity, near surface U, V, winds and surface pressure. In addition to the forcing files, the user also specifies parameters such as the spatial and temporal resolution, the land surface model, etc. LIS also allows users to initialize state variables, either by specifying a global uniform value or taken from a restart file produced by a previous run. Please refer to the LDAS source code documentation (http://lis.gsfc.nasa.gov/docs/LDAS-Doc/ldas2/) for a detailed description of the input/output routines corresponding to each file. The output from the land models translates to variables such as soil moisture, surface runoffs, canopy transpiration, etc. A list of the LIS variables passed between the modules, following the ALMA standard, is presented in Appendix A.

The atmospheric forcing data, fetched from various locations on the Internet, need to be fed into the compute nodes at regular intervals. The total data volume is estimated to be 279 MB/day. We designate one of the IO nodes to fetch and pre-process the data, then the processed atmospheric forcing data are distributed to the compute nodes' parallel processes by one or more master processors running on the IO nodes. The compute nodes can also access the forcing data through NFS system.

Table 1 describes the atmospheric forcing data to be used by LIS, and the estimated data volume. The forcing data, depending on their origination, have different spatial resolution and temporal intervals, and the estimation shows LIS will deal with an incoming data flux of approximately 279MB.

Table 1: Atmospheric forcing data description and volume estimation

Dataset	*Description*	*Desired resolution*	*Native format*	*Approx size*	*Update frequency*
GDAS forcing data	The Global Data Assimilation System (GDAS) is the global, operational weather forecast model of NCEP(Derber et al 1991). LDAS makes use of GDAS 0, 0.3, and, as needed, 6 (hour) forecasts, which are produced at 6 hour intervals	Native T170, ~0.7deg	GRIB	50M/day (3.2M X 4 X4)	Every 6 hours
GEOS forcing data	Obtained from GSFC's Goddard Earth Observing System Data Assimilation System (GEOS) (Pfaendtner et al. 1995) version 4.3 that supports level-4 product generation for the NASA Terra satellite (Atlas and Lucchesi 2000).	1 deg	Binary	25M/day	Every 3 hours
AGRMET SW flux data	LDAS estimates global, downward shortwave and longwave radiation fluxes using a procedure from the Air Force Weather Agency's (AFWA) Agricultural Meteorology modeling system (AGRMET). It utilizes the AFWA Real Time Nephanalysis (RTNEPH) 3-hourly cloud maps (Hamill et al. 1992), and the AFWA daily snow depth (SNODEP) maps (Kopp and Kiess 1996) to calculate surface downwelling shortwave radiation using the algorithms of Shapiro (1987)	~48km	Binary	48M/day	Every 1 hour
AGRMET LW flux data			Binary	144M/day	Every 1 hour
NRL Precipitation data	Near-real time satellite-derived precipitation data is obtained from the U. S. Naval Research Laboratory (NRL). NRL produces precipitation fields based on both geostationary satellite infrared (IR) cloud top temperature measurements and microwave observation techniques (Turk et al. 2000)	1/4 degree	Binary	12M/day	Every 6 hours
Total data input flux				**279M/day**	

The land surface parameter data include the vegetation classification, land mask, soil properties, leaf area index (LAI), etc., with a volume of about 136 GB. Since these data will not be updated frequently, we will put a copy of these data on each compute node's local disk to reduce network traffic. Currently the bulk of the data are saved as ASCII data, and we will convert the data into binary format to allow all the static data to fit on each compute node's 80 GB disk. Table 2 describes the land surface parameter data and gives the volume estimation.

Table 2: Land surface parameter data description and volume estimation

Dataset	*Description*	*Desired resolution*	*Native format*	*Approx size*	*Update frequency*
UMD Vegetation classification map	This file lists the frequency with which of each of the 14 vegetation types occurs in each of the 0.25 degree LDAS grid boxes. See http://ldas.gsfc.nasa.gov/GLDAS/VEG/GLDASveg.shtml for a detailed description.	1km X 1km	ASCII	65G	Static
UMD Land mask	This ascii file contains the LDAS unified land/sea mask.See http://ldas.gsfc.nasa.gov/GLDAS/VEG/GLDASveg.shtml for a detailed description.	1km X 1km	ASCII	18G	Static
Soil classification map	The soil parameter maps used in LDAS were derived from the global soils dataset of Reynolds et al. (1999). That dataset includes the percentages of sand, silt, and clay, among other fields, and is based on the United Nations Food and Agriculture Organization (FAO) Soil Map of the World linked to a global database of over 1300 soil pedons. The LDAS soil color map was interpolated from a 2 x 2.5 degree global map produced by NCAR0.01.bin	1km X 1km	ASCII	20G	Static
Soil color map		1km X 1km	Binary	2G	Static
Sand fraction file		1km X 1km	Binary	6G	Static
Clay fraction map		1km X 1km	Binary	6G	Static
Leaf area index (LAI)	This was generated using three information sources: (1) an 8km resolution time series of LAI, which was derived by scientists at Boston University (Myneni et al. 1997) from AVHRR measurements of normalized difference Avegetation index (NDVI) and other satellite observations.(2) A climatology based on the 8km time series and (3) the 1km UMD vegetation type classification.	1km X 1km	Binary	1M	Static
AVHRR-derived LAI climatology		1km X 1km	Binary	5G	Static
Static file size				**136G**	

The output data, produced by the distributed compute nodes in parallel, will be collected, assembled, and stored on the IO nodes too, and served to users via a GrADS-DODS Server and a LAS (Live Access Server) server running on one of the IO nodes. Since it is not feasible to store the output in a single file (200 GB/day, see below), we want to distribute the data across all the IO nodes. To keep the huge output data volume manageable, we designed a storage scheme that will distribute the land surface variables in the output data across the IO nodes. Since there are 40-48 variables in the output data, with some of them having multiple levels, we can let each IO node to store the global data of only 6 or so of the output variables. So on average, the I/O traffic is segregated and each IO node is only taking 1/8 of the total data traffic, and the subsequent operations by the GrADS-DODS server and the LAS server are greatly simplified. Table 3 gives the output data volume estimate for various spatial resolutions. For the finest target resolution (1km by 1km), the output data flux is estimated to be 200GB/day.

Table 3: LIS output volume estimation

Dataset	*Description*	*Desired resolution*	*Native format*	*Approx size*	*Update frequency*
CLM output data	LIS output data of ~37 variables	1km X 1km	GRIB	200G/day	every hour
		5km X 5km		8G/day	
		1/8 X 1/8 deg		0.9G/day	
		1/4 X 1/4 deg		0.2G/day	
Total data output flux				**200G/day**	

2.4 Network Data Traffic Estimation

As shown in Table 3 above, the total output data volume (200GB/day) produced by the cluster is much larger than the input data volume (279MB/day), so the network traffic is dominated by the upstream traffic from the compute nodes to the IO nodes, where the output data are stored. The data will travel trough two network links: link A -- a compute node to a fast Ethernet switch port; link B -- the switch's gigabit port to an IO node. Figure 2 shows the network traffic between the two network links. Following is the worksheet for the estimation of the traffic at these two links:

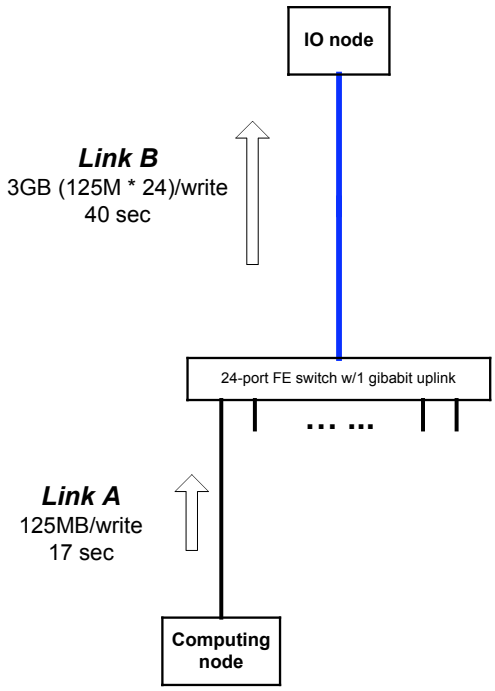

Figure 2: Network traffic estimation within the cluster. The traffic is dominated by the output data flow from the compute nodes to the IO nodes. The output data will go through two network links: Link A, from a compute node to an Ethernet switch; Link B, from the switch's gigabit port to an IO node.

Worst case scenario assumption: all the compute nodes are writing the output data to the IO nodes at the same time; effective bandwidth is 60% of the Ethernet wire bandwidth. Link A traffic:

Data volume each compute node will produce:
`200GB/day * (1/192) ~ 1GB/day`

Frequency each compute node writes output data to an IO node:
`every 3 simulation hours`

Total writes a compute node has:
`8 per simulation day`

Data volume each write per compute node:
`1GB/day * (1/8) = 125MB/write`

Time taken for the data to travel over link A:
`125MB*8/(100M*60%)= 17 sec`

Link B traffic:
In average, the number of compute nodes each IO node receives data from:
`24`

Total data volume each IO node receives per write:
`125MB * 24 = 3GB`

Time taken for the data to travel over link B:
`3GB*8 /(1G*60%) = 40 sec`

In summary, in the worst case scenario, it takes only 40 seconds for the 3-hour simulation data to be transferred from the compute nodes to the IO nodes. In reality, the data traffic will be much spread in time, so the network is even less likely to be a bottleneck.

To support this estimate, Figure 3 below shows benchmark measurements of the cluster's effective bandwidth over two kinds of network links: a compute node to an IO node, and a compute node to a compute node. The benchmark tests were performed over TCP protocol, over which the MPI messages are passed. The measurements show that for large data blocks (> 3000B), the effective bandwidth for both links are well above 60% (60Mbps) of the Ethernet wire bandwidth (100Mbps). For LIS' MPI jobs, we will tune the message buffer size to take advantage of the large data block performance.

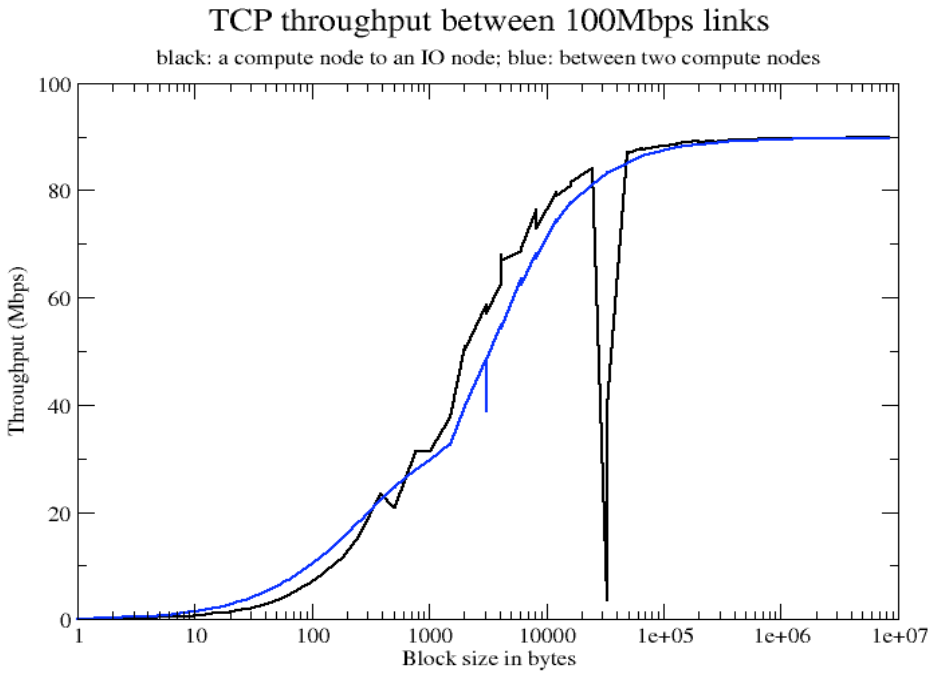

Figure 3: : Network bandwidth benchmark tests. Black curve: TCP throughput between a compute node and an IO node. Blue curve: throughput between two compute nodes. The tests were done using the NetPIPE package (http://www.scl.ameslab.gov/netpipe/).

3. Input Data

3.1 Atmospheric forcing data

The atmospheric forcing data are centered around two independent atmospheric simulation/assimilation models, the Global Data Assimilation System (GDAS) by NCEP, and the Goddard Earth Observing System Data Assimilation System (GEOS) by GSFC. In addition, a selected set of real-time, operational atmospheric observations, including radiation and precipitation, are incorporated, whenever available, into either the GDAS or GEOS forcing, to reduce any possible model biases.

Figure 4 and 5 illustrate the process to acquire the GDAS and GEOS forcing data, respectively. The process to fetch, store, pre-process and distribute these two sets of forcing data, as well as the other observational data, is similar. The original forcing data are first downloaded from the Internet source sites to LIS local disks, automatically by cron jobs at the regular intervals specific to each dataset. The local copies are then processed to select only the data to be used by LIS, and converted to the desired format. These processed data are then imported by LIS and distributed to the land surface models.

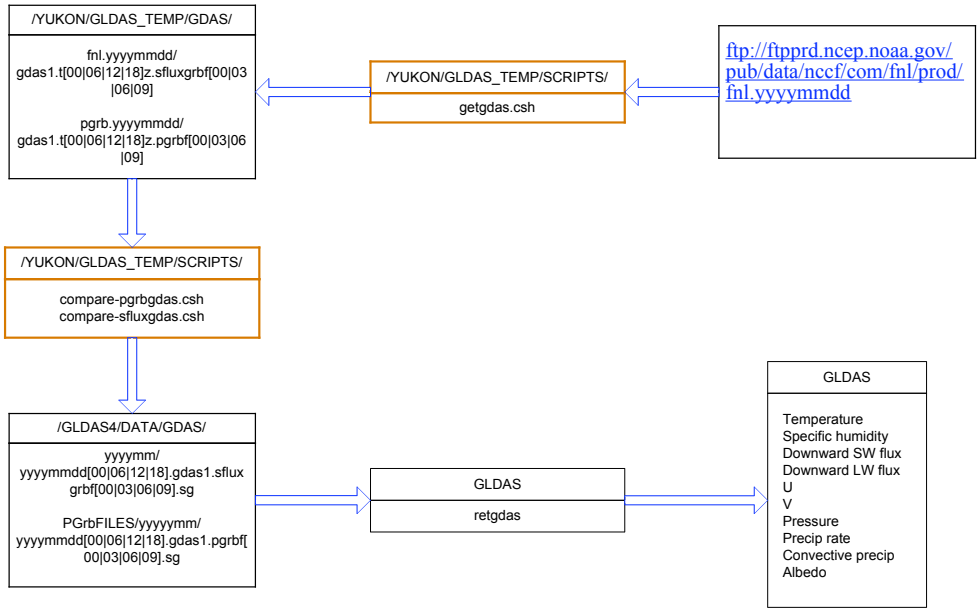

Figure 4: Acquisition of GDAS forcing data by LIS. The forcing data is fetched periodically from NCEP's operational archive, temporarily stored on the local disks, processed by various programs, and finally fed to the land surface models with ten variables**.**

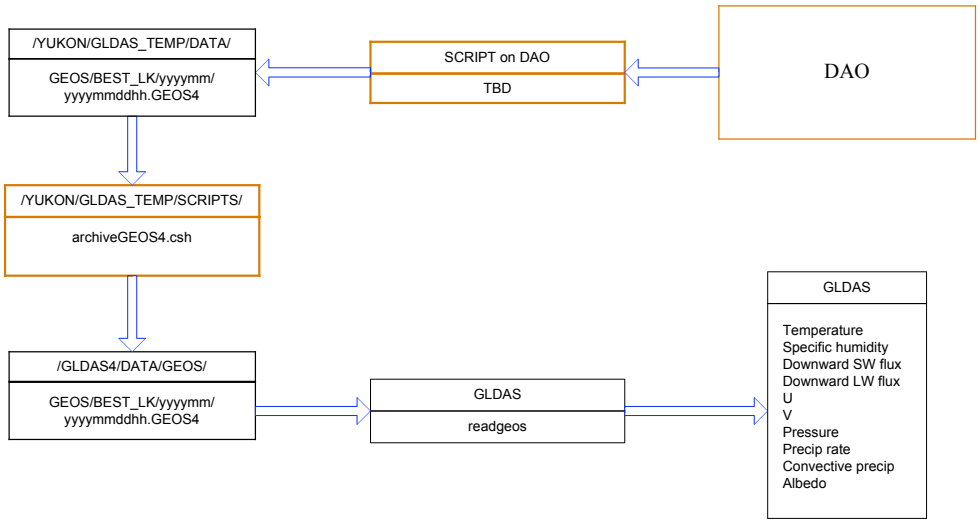

Figure 5: Acquisition of GEOS forcing data by LIS. The forcing data is pushed periodically from DAO's operational archive to the local disks, temporarily stored there, processed by various programs, and finally fed to the land surface models with ten variables.

3.2 Land surface parameter data

The land surface parameters, as listed in Table 2, are collected from a diverse set of sources. Most of these parameters are not time-dependent, so the acquisition and processing of them are a one-time operation. However, as newer land parameter data are constantly made available, the process of obtaining these parameters, and the programs used for this purpose, will be modified from time to time. Figure 6 shows the process for

the land surface parameter data collection. The source data are in disparate formats and resolutions, so a unique program is developed to handle each data source.

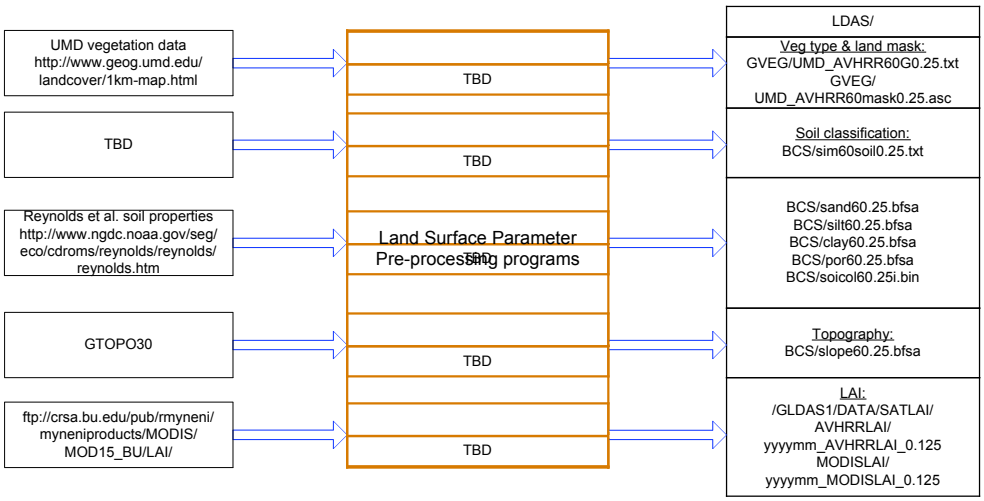

Figure 6: Acquisition of land surface parameters. This process features a diverse collection of data sources.

4. Output Data and Connection to the User Interface

LIS output data will be archived locally on the IO nodes' RAID disks. We will keep as much historical data as permitted by the storage capacity. However, the most recent data will be given the top storage and retrieval priority, and historical data will be removed if needed to make room for newest datasets.

To ensure flexible and easy access to LIS' huge amount of output data, we will deploy two web-based data supplying engines, the GrADS-DOS server and the Live Access Server (LAS), in addition to the conventional FTP server. Figure 7 shows the architecture of these two engines, and their interface with the output data archive. Together, they will provide users with the capabilities to perform data searching, data analysis, subsetting, visualization, format conversion and re-projection.

The GrADS-DODS server will provide data access via DODS-protocol to DODS clients. The GrADS-DODS server uses a typical client—server architecture to communicate with the DODS clients. The communication protocol between a client and a server is HTTP. A GrADS-DODS server has the following components: Java servlets contained in the Tomcat servlet container, to handle the client requests and server replies via HTTP protocol; DODS server APIs, to parse the DODS requests and package output data; interface code, to translate the DODS requests into GrADS calls; and finally, GrADS running in batch mode, to actually process the requests, and perform data-retrieving, sub-setting and processing on the server side.

The LAS server provides a web interface for users to search a data catalog, to visualize the data interactively, and to download the data in various formats. LAS uses perl scripts

to retrieve the metadata from the LIS output files, and save the metadata in a SQL database system, MySQL. The metadata catalog will be presented as a selectable and searchable web page. A user can pick a dataset, or a spatial and temporal subset of it, and interactively generate images on the fly using the graphics engine Ferret via the standard CGI mechanism (see http://ferret.pmel.noaa.gov/Ferret/ for more technical details). An ongoing effort is to integrate the popular GrADS into LAS as another graphics engine, so the users can have the option to choose the image engine they prefer.

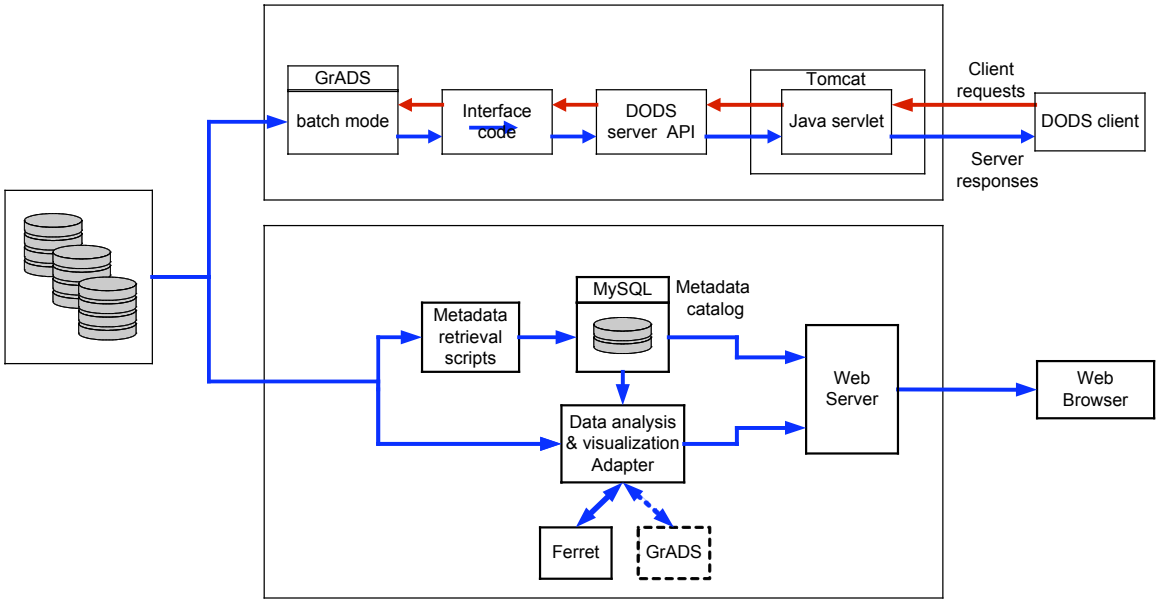

Figure 7: Architecture of GrADS-DODS (GDS) server and Live Access Server (LAS), and their connections to the LIS output data. GDS and LAS were developed by COLA (http://grads.iges.org/grads/gds/) and NOAA (http://ferret.wrc.noaa.gov/Ferret/LAS/ferret_LAS.html), respectively.

5. ALMA and ESMF Compliance

5.1 ALMA Standard Compliance

The ALMA standard specifies a unified scheme for the names, units and sign conventions of land surface and atmospheric forcing variables, as well as the numerical format for the data files holding there variables. Compliance with ALMA standard will facilitate data exchange between different land surface and atmospheric models. LIS is committed to follow the ALMA standard, and the data flow will follow ALMA's specification. Appendix A lists the ALMA variables used by LIS. The *Interoperability Design Document* gives more detailed description for the ALMA compliance design and implementation.

5.2 ESMF Compliance

Based on component technologies, ESMF defines the architecture and interfaces of numerical models for Earth system simulations. It will also provide a standardized collection of utilities to make the implementation of the component interfaces easier. This framework will standardize the coupling between various Earth system models, and facilitate reuse and interoperability of the modeling code.

Guided by the ESMF architecture specification, we designed the software architecture of LIS data management as shown in Figure 8 below. The land surface parameter data are encapsulated inside the land models as part of their intrinsic data structure. LIS get the atmospheric forcing data, on the other hand, in a way as if it gets the data from a ESMF-compliant atmospheric model. We will wrap the data supplying mechanism for the forcing data with the ESMF atmospheric model component, and supply the forcing data with the standard interfaces. LIS will invoke the standard ESMF mechanism to acquire the forcing data via the ESMF coupler component. LIS will also provide ESMF interfaces to supply its output data (Export_state) to other ESMF-compliant models as their input.

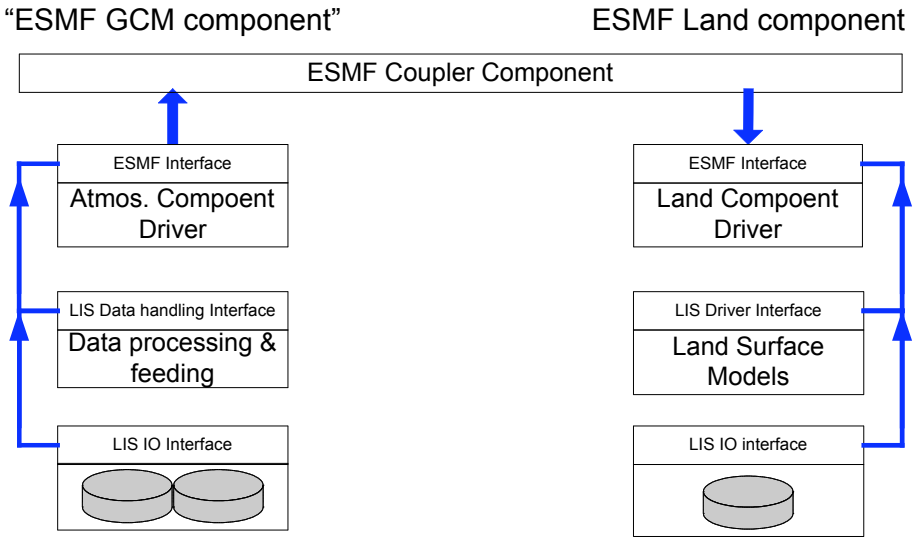

Figure 8: ESMF-compliant software architecture for LIS data management. The land surface parameter data are encapsulated inside the land surface models. The atmospheric forcing data are wrapped with a ESMF-compliant atmospheric component, and the data are supplied to LIS via the ESMF coupler component.

Appendix A

List of LIS variables passed between the modules (Following ALMA Convention)

```
nswrs Net Surface Shortwave Radiation (W/m2)
nlwrs Net Surface Longwave Radiation (W/m2)
lhtfl Latent Heat Flux (W/m2)
shtfl Sensible Heat Flux (W/m2)
gflux Ground Heat Flux (W/m2)
snohf Snow Phase Change Heat Flux (W/m2)
dswrf Downward Surface Shortwave Radiation (W/m2)
dlwrf Downward Surface Longwave Radiation (W/m2)
asnow Snowfall (kg/m2)
arain Rainfall (kg/m2)
evp Total Evaporation (kg/m2)
ssrun Surface Runoff (kg/m2)
bgrun Subsurface Runoff (kg/m2)
snom Snowmelt (kg/m2)
snowt Snow Temperature (K)
vegt Canopy Temperature (K)
baret Bare Soil Surface Temperature (K)
avsft Average Surface Temperature (K)
radt Effective Radiative Surface Temperature (K)
albdo Surface Albedo, All Wavelengths (%)
weasd Snowpack Water Equivalent (kg/m2)
cwat Plant Canopy Surface Water Storage (kg/m2)
soilmc Total Column Soil Moisture (kg/m2)
soilmr Root Zone Soil Moisture (kg/m2)
soilmt1 Top 1-meter Soil Moisture (kg/m2)
mstavc Total Soil Column Wetness (%)
mstavr Root Zone Wetness (%)
evcw Canopy Surface Water Evaporation (W/m2)
trans Canopy Transpiration (W/m2)
evbs Bare Soil Evaporation (W/m2)
sbsno Snow Evaporation (W/m2)
pevpr Potential Evaporation (W/m2)
acond Aerodynamic Conductance (m/s)
lai Leaf Area Index
snod Snow Depth (m)
snoc Snow Cover (%)
salbd Snow Albedo (%)
tmp2m Two Meter Temperature (K)
humid Two Meter Humidity (kg/kg)
uwind Ten Meter U Wind (m/s)
vwind Ten Meter V Wind (m/s)
sfcprs Surface Pressure (mb)
soilt Soil Temperature (K)
soilm Soil Moisture (kg/m2)
lsoil Liquid Soil Moisture (kg/m2)
```